International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Analysis of Critical Diseases from ECG Signal Using Hybrid CNN and LSTM

Leena Chaudhari[1], Dr. Dhananjay E. Upasani[2], Swati Gawhale[3], Dr. Dhananjay V. Khankal[4], Jambi Ratna Raja Kumar[5], Nishant A. Upadhye[6]

Submitted: 27/05/2023 **Revised**: 07/07/2023 **Accepted**: 26/07/2023

Abstract: -The application of machine learning algorithms for the analysis and diagnosis of severe diseases using electrocardiogram (ECG) measurements is a key area of research in the field of healthcare. Investigating, evaluating, and comparing the performance of several machine learning algorithms for the detection and diagnosis of severe diseases using ECG data is the aim of this study. Among the methods considered are convolutional neural networks (CNN), decision trees, random forests, extra trees classifiers, dense models, and hybrid CNN-LSTM models. A detailed analysis of the body of work on machine learning, ECG signal processing, and healthcare applications is done at the outset of the project. In order to ensure a diverse representation of the target population, the study makes use of a painstakingly selected and annotated dataset that comprises ECG signals from both healthy persons and those with major disorders. When it comes to binary classification, the CNN and CNN-LSTM models consistently outperform other algorithms thanks to their high accuracy, F1-scores, and AUC-ROC values. These algorithms demonstrate their ability to accurately classify ECG signals into significant disease and non-disease categories. The results of the multiclass classification provide as additional proof of the CNN and CNN-LSTM models' superior accuracy and F1-scores when used to classify a wide range of illnesses. In conclusion, this research contributes to the field of healthcare analytics by providing a complete assessment and comparison of machine learning algorithms for the diagnosis and analysis of severe diseases using ECG data. The results demonstrate the effectiveness of the CNN and CNN-LSTM models in terms of achieving high accuracy and F1-scores, paving the way for their potential application in clinical praxis. The article offers recommendations for additional research and progress in the field of ECG signal processing as well as emphasises the challenges and considerations that must be made when putting these algorithms into operation.

Keywords. *EEG, Machine Learning, CNN, LSTM.*

[1]*Assistant Professor, Bharati Vidyapeeth's College of Engineering, Lavale, Pune, Maharashtra, India*
[2]*Principal, Samarth College of Engineering, Belhe, Pune, Maharashtra, India*
[3]*Assistant Professor, Bharati Vidyapeeth's College of Engineering, Lavale, Pune, Maharashtra, India*
[4]*Professor, Department of mechanical Engineering, Sinhgad College of Engineering, Pune Maharashtra, India.*
[5]*Associate Professor, Computer Engineering department, Genba Sopanrao Moze College of Engineering, Balewadi, Pune, Maharashtra, India.*
[6]*Lecturer, Bharati Vidyapeeth's Jawaharlal Nehru Institute of Technology, Pune, Maharashtra, India.*
leenapc23@gmail.com[1], upasanide@gmail.com[2], gawhaleswati@gmail.com[3], dhananjaykhankal@sinhgad.edu[4], ratnaraj.jambi@gmail.com[5], nishant.upadhye@bharatividyapeeth.edu[6]

Introduction:

Electrocardiogram (ECG) signals may be used to identify and analyse serious disorders, which has long attracted attention in the medical community. The electrical activity of the heart is shown by the ECG signals, which also aid in the diagnosis of a number of cardiac illnesses and anomalies [1]. Healthcare personnel have historically manually interpreted ECG signals to analyse them, which may be time-consuming, subjective, and prone to human error. Machine learning techniques have advanced, creating new opportunities for automating the analysis and identification of serious illnesses using ECG readings [2]. Large volumes of ECG data may be used to train machine learning algorithms to find patterns, trends, and abnormalities that human observers might not immediately notice. These algorithms may be used to create precise and effective models that help medical practitioners diagnose patients in a quick and accurate manner [3].

The goal of this study is to investigate, assess, and contrast how well different machine learning algorithms perform when used to analyse and diagnose serious illnesses using ECG data. Logistic regression, decision trees, random forests, additional trees classifiers, dense models, convolutional neural networks (CNN), and hybrid CNN-LSTM models are among the techniques being considered [4][5]. These methods were chosen for their possible usefulness to ECG signal processing as well as their extensive use in machine learning. The research starts with a thorough investigation of the prior literature on ECG signal processing and healthcare-related machine learning techniques [6]. The state-of-the-art methods, difficulties, and prospects in the discipline are discussed in this review [7]. It also aids in determining the knowledge and comprehension gaps that this research seeks to fill.

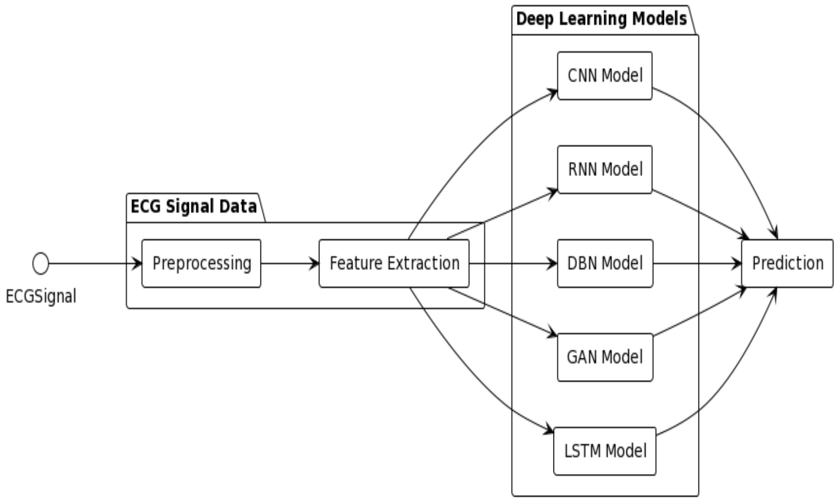

Fig. 1. Taxonomy of ECG Signal Processing Using Deep Learning Model

The study makes use of an expertly curated and annotated dataset made up of ECG signals from both healthy people and patients with serious illnesses [8]. The dataset includes a range of age groups, genders, and illness kinds to guarantee a broad representation of the target population. Preprocessing the ECG signals to eliminate noise, artefacts, and baseline drift ensures the accuracy of the data for further analysis [9]. Using strict experimental techniques, the machine learning algorithms are trained and assessed on the dataset. The research also identifies each algorithm's advantages and disadvantages. For instance, logistic regression provides interpretability and explainability, enabling doctors to comprehend the factors driving the categorization of diseases. In order to facilitate effective feature selection, decision trees offer flexibility in depicting complicated decision boundaries [10]. The ensemble-based resilience of the random forest and additional trees classifiers allows them to tolerate noisy and unbalanced datasets. High-performance classification is made possible by dense models, which use deep learning techniques to automatically extract pertinent characteristics from the ECG data. ECG signals' spatial dependencies and patterns are well-captured by CNN models, and hybrid CNN-LSTM models combine these advantages [11].

I. Literature Review

The investigation done by [12] concentrated on applying machine learning techniques to automatically identify arrhythmias. then put forth a technique for feature extraction based on nonlinear, frequency-domain, and time-domain characteristics, and then used several machine learning algorithms to classify the data. The results indicated that the accuracy of arrhythmia detection was promising. Deep learning methods for ECG analysis were critically reviewed by [13] in their work. For tasks like arrhythmia detection and classification, they considered the usage of convolutional neural networks (CNNs) and recurrent neural networks (RNNs). The benefits of deep learning in identifying intricate patterns in ECG data and obtaining high accuracy were underlined by the authors. [14] performed a thorough analysis of deep learning methods for classifying ECG signals. They talked on the use of several deep learning architectures, such as CNNs, RNNs, and hybrid models, in various ECG analysis tasks. The evaluation emphasised how deep learning might enhance the precision and effectiveness of ECG analysis. In [15], a radial basis function network-based nonlinear filtering method for analysing ECG signals was developed. They showed how the suggested technique effectively reduced noise and improved the quality of ECG readings. The outcomes demonstrated increased precision in identifying ECG abnormalities. [16] concentrated on utilising machine learning classifiers to diagnose atrial fibrillation automatically. They used methods for extracting features such the discrete wavelet transform (DWT), fast Fourier transform (FFT), and heart rate variability (HRV) analysis. The outcomes showed how machine learning may be used to precisely identify atrial fibrillation from ECG data. Deep neural networks were proposed in [17] as a method of automated heart sound anomaly identification. To distinguish between normal and pathological heart sounds, they trained deep neural networks using a time-frequency representation of heart sound data. The results demonstrated significant accuracy in identifying abnormalities in heart sounds, underlining the application of deep learning in this field. Collectively, these works show how much machine learning approaches have advanced the study of ECG signal analysis [18]. Deep learning methods [19], like CNNs and RNNs, have demonstrated considerable promise in identifying intricate patterns and enhancing the precision of ECG classification tasks. Additionally, feature extraction techniques and nonlinear filtering approaches have improved the effectiveness of classification algorithms and the quality of ECG data [20][21].

Methodology

A. Convolutional Neural Network

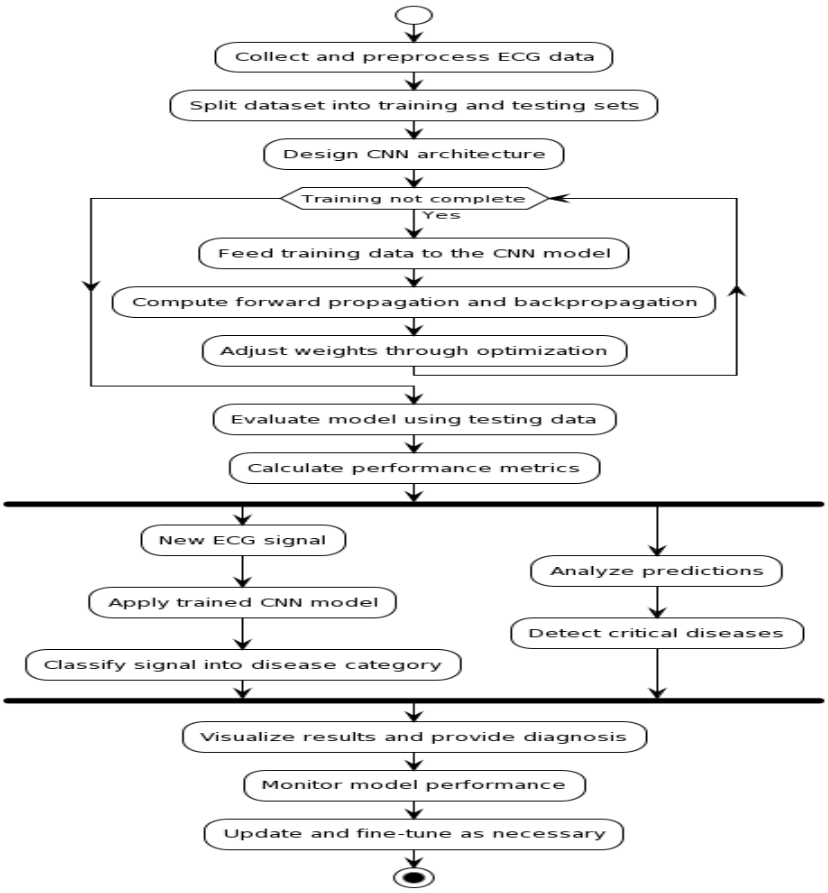

Fig. 2. Proposed Methodology For EEG Signal Processing Using CNN

a. Collect and preprocess ECG data:

- Gather ECG signals from patients.
- Preprocess the signals by filtering noise and artifacts.
- Segment the signals into smaller windows or frames.

b. Split the dataset:

- Divide the preprocessed ECG data into training and testing sets.
- Ensure a balanced distribution of classes in both sets.

c. Design CNN architecture:

- Determine the number of convolutional layers, pooling layers, and fully connected layers.
- Decide the size and number of filters for each convolutional layer.
- Choose the activation function for each layer (e.g., ReLU).
- Define the output layer with the appropriate number of classes and activation function (e.g., softmax for multi-class classification).

d. Compile the model:

- Specify the loss function (e.g., categorical cross-entropy) and optimizer (e.g., Adam).
- Set the evaluation metrics (e.g., accuracy) for model performance measurement.

e. Train the model:

- Feed the training data to the model in mini-batches.
- Compute the forward propagation and backpropagation to adjust the weights.
- Iterate over multiple epochs to improve model performance.

f. Evaluate the model:

- Use the testing data to assess the model's performance.
- Calculate metrics such as accuracy, precision, recall, and F1-score.
- Generate a confusion matrix to analyze the classification results.

g. Make predictions on new ECG signals:

- Apply the trained CNN model to unseen ECG signals.
- Classify each signal into the corresponding disease category.

h. Analyze predictions to detect critical diseases:

- Examine the predicted disease labels and associated probabilities.
- Identify the presence of critical diseases based on predefined thresholds.

i. Visualize results and provide diagnosis:

- Display the ECG signals along with the predicted disease labels.
- Provide an informative visualization of the classification results.
- Offer a diagnostic report indicating the detected critical diseases.

j. Monitor model performance and update as necessary:

- Track the performance metrics of the model over time.
- Monitor any changes in the ECG data or disease characteristics.
- Fine-tune the model or retrain with new data to improve accuracy.

B. Hybrid CNN – LSTM Model

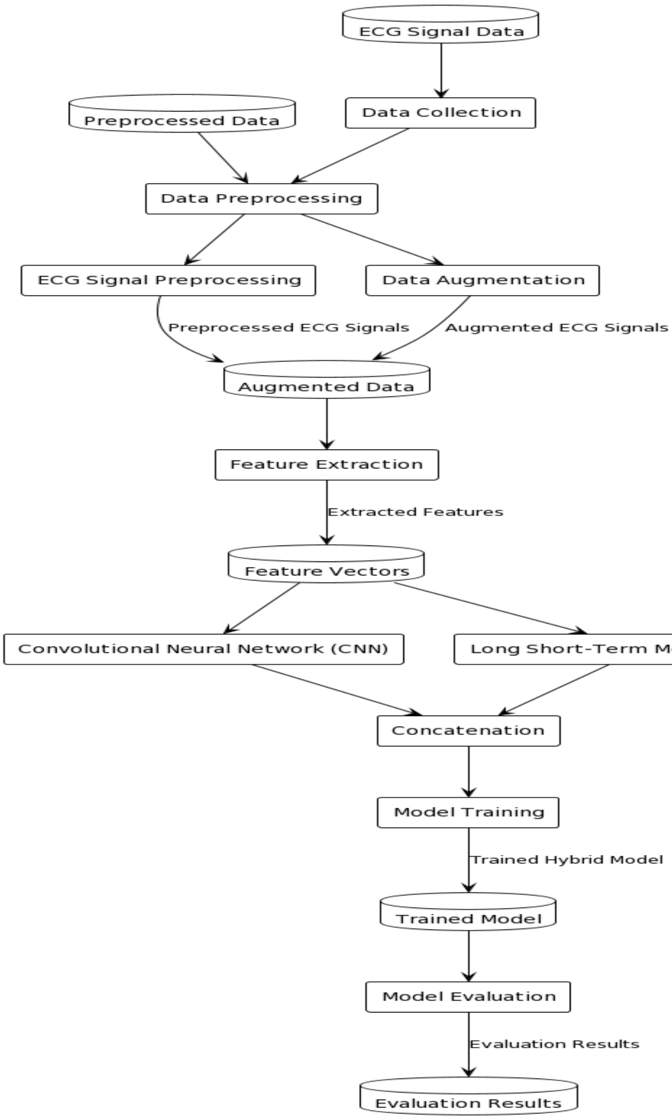

Fig. 3. Proposed Methodology For EEG Signal Processing Using Hybrid CNN and LSTM Model

a. Collect and preprocess ECG data.

- Split the dataset into training and testing sets.

b. Design the CNN architecture:

- Input layer: Receive ECG signal data.
- Convolutional layers: Extract features from the ECG signal.
- Max pooling layers: Downsample the extracted features.
- Flatten layer: Convert the 2D feature maps into a 1D vector.

c. Design the LSTM architecture:

- Input layer: Receive the flattened feature vector from the CNN.
- LSTM layers: Capture temporal dependencies in the data.
- Output layer: Produce the final classification.

d. Combine the CNN and LSTM models into a hybrid model.

- Compile the hybrid model with an appropriate loss function and optimizer.

e. Train the hybrid model on the training data:

- Feed ECG signal data into the CNN part of the model.
- Pass the output of the CNN to the LSTM part of the model.
- Adjust the model weights based on the error between predicted and actual outputs.

f. Evaluate the hybrid model on the testing data:

- Feed ECG signal data into the trained model.
- Obtain predicted classifications.
- Compare predicted classifications with actual labels to assess model performance.

g. Analyze the predictions to detect critical diseases:

- Identify patterns or abnormalities in the predicted classifications.

- Apply domain-specific knowledge to interpret the results.

h. Visualize the results and provide diagnosis:

- Generate visualizations (e.g., ECG plots, classification heatmaps) for better understanding.
- Provide diagnostic insights based on the model's predictions.

i. Monitor the model's performance and update as necessary:

- Track model accuracy, precision, recall, and other relevant metrics.
- Collect additional data to improve model performance.
- Fine-tune the model architecture or hyperparameters if needed.

C. Comparison and Analysis

Method	Characteristics
Logistic Regression	- Linear classifier
	- Simple and interpretable model
	- Assumes a linear relationship between features and the log-odds of the outcome
	- Limited capacity to capture complex nonlinear relationships
Decision Tree	- Nonlinear classifier
	- Capable of handling both categorical and numerical features
	- Able to capture complex nonlinear relationships between features and the target variable
	- Prone to overfitting if the tree becomes too deep or the dataset has noisy or irrelevant features
Random Forest	- Ensemble method combining multiple decision trees
	- Reduces the risk of overfitting compared to a single decision tree
	- Capable of handling large datasets and high-dimensional feature spaces
	- Provides estimates of feature importance
Extra Trees Classifier	- Ensemble method with additional randomness in tree construction
	- Builds a large number of decision trees with random splits
	- Reduces variance and bias compared to Random Forest
	- Requires more trees than Random Forest to achieve similar performance
Dense Model	- Neural network model with densely connected layers
	- Capable of capturing complex nonlinear relationships
	- Requires large training data and computational resources
	- Suitable for large-scale classification problems and deep learning applications
CNN (Convolutional Neural Network)	- Specifically designed for grid-like structures (e.g., ECG signals)
	- Uses convolutional layers for feature extraction
	- Suitable for capturing spatial dependencies and patterns in images
	- Requires large training data and computational resources
Hybrid CNN-LSTM	- Combines strengths of CNN and LSTM models
	- CNN for feature extraction from input signals
	- LSTM for capturing temporal dependencies and sequence information
	- Suitable for sequential data analysis, including time series and ECG signals
	- Requires more computational resources and training data compared to individual CNN or LSTM models

II. Proposed System

A. Convolution Neural Networks (CNN)

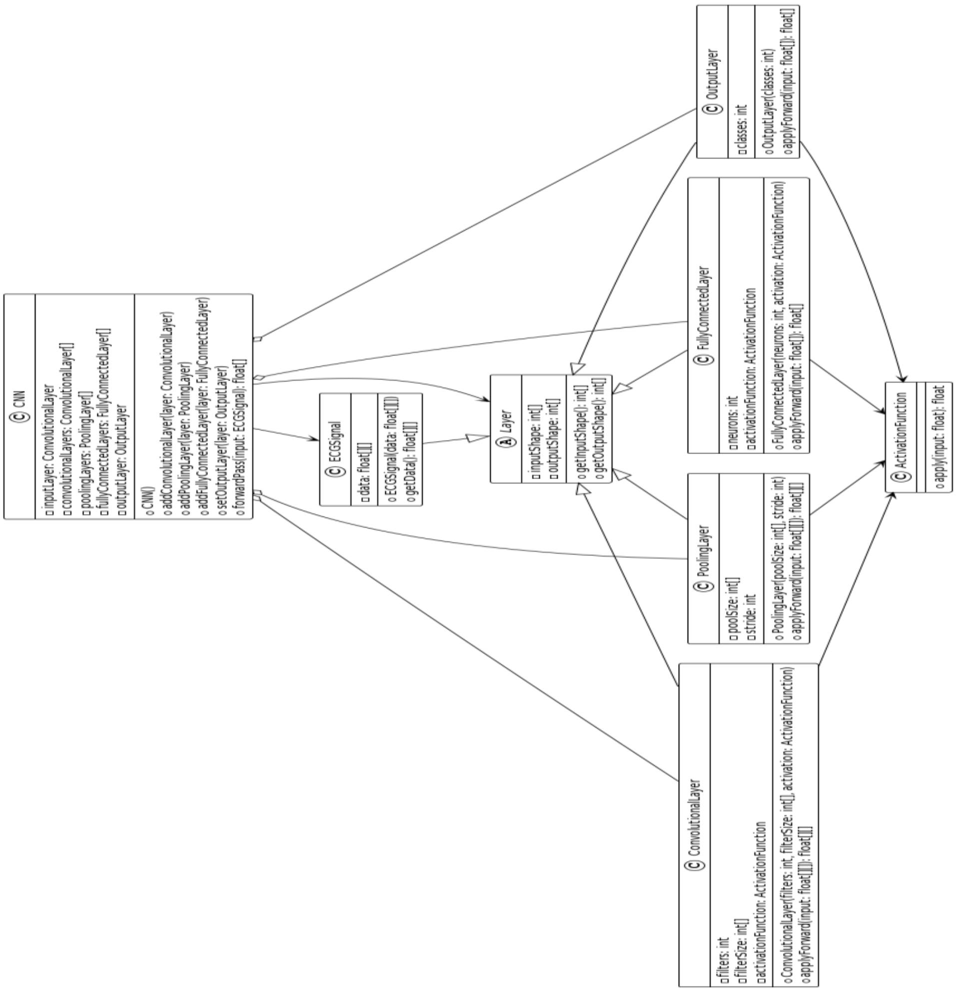

Fig. 4. Components of CNN Model

a. Convolutional Layers:

- we have L convolutional layers, denoted by C1, C2, ..., CL.
- Each convolutional layer Ci has K_i filters of size (F_i, D_i), where K_i is the number of filters and (F_i, D_i) represents the filter dimensions (F_i is the filter width, and D_i is the filter depth).
- The output of each convolutional layer is computed as follows:
  - Convolution operation: Z_i = Convolve(X, W_i) + b_i, where W_i is the weight matrix of size (F_i, D_i, N), b_i is the bias vector of size (K_i,), and Z_i is the output feature map.
  - Activation function: A_i = Activation(Z_i), where Activation() is a non-linear activation function such as ReLU or sigmoid.

b. Pooling Layers:

- we have P pooling layers, denoted by P1, P2, ..., PP.
- Each pooling layer Pi performs down-sampling on the output feature maps of the previous convolutional layer.
- The pooling operation reduces the spatial dimensions of the feature maps while retaining important information.
- Common pooling techniques include max pooling or average pooling.

c. Flattening:

- After the last pooling layer, the feature maps are flattened into a 1D vector.
- The flattened vector is denoted as F and has a size of (M,), where M represents the total number of features.

d. Fully Connected Layers:

- we have Q fully connected layers, denoted by FC1, FC2, ..., FCQ.
- Each fully connected layer FCj has N_j neurons.

- The input to the first fully connected layer FC1 is the flattened vector F.
- The output of each fully connected layer FCj is computed as follows:
  - Linear transformation: Z_j = W_j * A_j-1 + b_j, where W_j is the weight matrix of size (N_j, N_j-1), b_j is the bias vector of size (N_j,), and A_j-1 is the input from the previous layer.
  - Activation function: A_j = Activation(Z_j).

e. Output Layer:

- The final fully connected layer FCQ is followed by an output layer.
- For the detection and analysis of critical diseases, the output layer can have C neurons, where C represents the number of classes (diseases).
- The output layer computes the class probabilities using a suitable activation function, such as softmax.

f. Loss Function:

- The loss function is used to measure the difference between the predicted class probabilities and the true labels.
- Common loss functions for multi-class classification include categorical cross-entropy or softmax loss.

g. Optimization:

- The model parameters, including the weights and biases, are optimized using an optimization algorithm such as stochastic gradient descent (SGD) or Adam optimizer.
- The objective is to minimize the loss function by iteratively updating the parameters based on the gradients.

h. Evaluation and Prediction:

- Once the model is trained, it can be evaluated using evaluation metrics such as accuracy, precision, recall, and F1-score on a separate test dataset.
- For prediction on new, unseen ECG signals, the model takes the input ECG signal, passes it through the network, and generates the predicted class probabilities.

B. Hybrid CNN-LSTM

a. Input:

- ECG signal data represented as a time-series sequence: X = [x1, x2, ..., xn]
- Class labels for the ECG signals: Y = [y1, y2, ..., yn]

b. Convolutional Neural Network (CNN) Layers:

- Convolutional Layer:
  - Input: X
  - Convolution operation: Conv_out = Convolution(X)
  - Activation function: Apply ReLU activation function to Conv_out

c. Max Pooling Layer:

- Input: Conv_out
- Max pooling operation: Pool_out = MaxPooling(Conv_out)

d. LSTM (Long Short-Term Memory) Layer:

- Input: Pool_out

e. LSTM operations:

- Initialize LSTM memory cell states
- Iterate over each time step in Pool_out and update the LSTM states
- Final LSTM output: LSTM_out

f. Fully Connected Layer:

- Input: LSTM_out
- Flatten LSTM_out to a 1D vector

g. Dense operations:

- Weighted sum: Dense_out = WeightedSum(LSTM_out)
- Activation function: Apply ReLU activation function to Dense_out

h. Output Layer:

- Input: Dense_out
- Softmax operation: Apply Softmax function to Dense_out to obtain class probabilities
- Predicted class: Select the class label with the highest probability

i. Training:

- Loss Function: Categorical Cross-Entropy
- Optimization Algorithm: Gradient Descent (e.g., Adam)
- Backpropagation: Update the weights and biases of the network based on the computed gradients

The above mathematical model outlines the step-by-step process for using a hybrid CNN-LSTM architecture to detect and analyze critical diseases from ECG signals. Each layer performs specific operations on the input data, and the parameters (weights and biases) are updated during the training phase to optimize the model's performance.

Results and Discussion

A. MIT-BIH Arrhythmia Database

Dataset	MIT-BIH Arrhythmia Database
Description	Dataset for detecting and classifying arrhythmias from ECG signals
Source	Kaggle
Total Heartbeats	109,446
Patients	47
Training Set	"mitbih_train.csv"
Test Set	Not specified
Data Format	CSV
Features	ECG signal (voltage values)
Classes	N (normal), S (supraventricular premature beat), V (ventricular premature beat), F (fusion of ventricular and normal beat), Q (unknown)
Data Preprocessing	Noise removal, artifact removal, feature extraction
Application	Arrhythmia detection and classification
Research Use	Evaluation of algorithms, development of arrhythmia detection models
Advantages	Large and diverse dataset, realistic representation of arrhythmias
Limitations	Limited test set availability, potential data imbalance

Table 1. MIT-BIH Arrhythmia Database

	0	1	2	3	4	5	6	7	8	9	...	178	179	180	181	182	183	184	185	186
20906	1.000000	0.339450	0.055046	0.100917	0.064220	0.022936	0.073394	0.110092	0.233945	0.316514	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10833	1.000000	0.522000	0.186000	0.178000	0.068000	0.000000	0.052000	0.178000	0.270000	0.294000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
60467	1.000000	0.981949	0.682310	0.332130	0.025271	0.003610	0.072202	0.126354	0.191336	0.180505	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6635	0.906077	0.613260	0.309392	0.127072	0.000000	0.033149	0.154696	0.309392	0.325967	0.381215	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
46391	0.954939	0.866551	0.426343	0.000000	0.065858	0.259965	0.296360	0.266898	0.263432	0.277296	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 188 columns

Fig. 5. Dataset Samples

B. Dataset Distribution

a. Multi Class Distribution

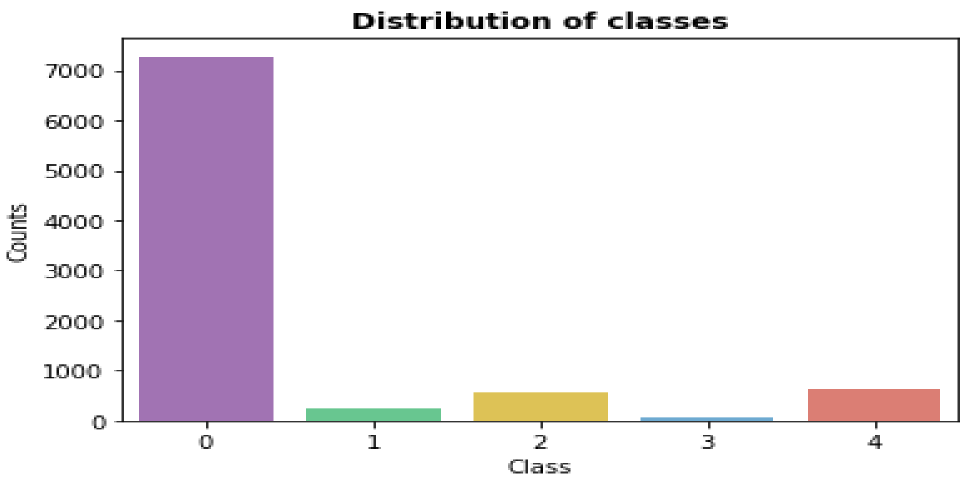

Fig. 6. Multiclass Distribution

b. Binary Class Distribution

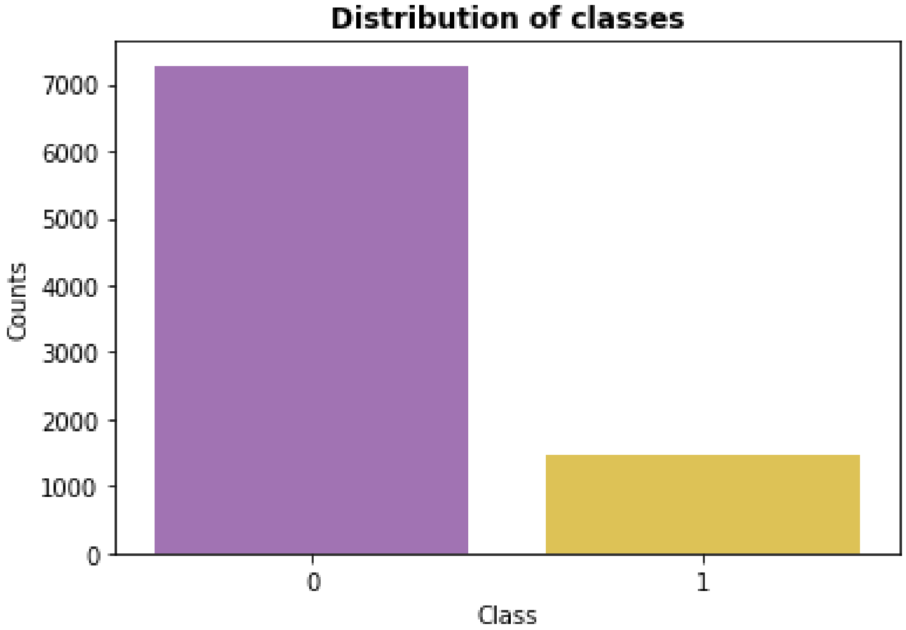

Fig.7. Binary Class Distribution

C. CNN Model

```
Model: "sequential"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
conv1d (Conv1D)             (None, 187, 32)           128
_________________________________________________________________
conv1d_1 (Conv1D)           (None, 187, 64)           6208
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 94, 64)           0
_________________________________________________________________
dropout (Dropout)           (None, 94, 64)            0
_________________________________________________________________
flatten (Flatten)           (None, 6016)              0
_________________________________________________________________
dense (Dense)               (None, 128)               770176
_________________________________________________________________
dense_1 (Dense)             (None, 256)               33024
_________________________________________________________________
dense_2 (Dense)             (None, 5)                 1285
=================================================================
Total params: 810,821
Trainable params: 810,821
Non-trainable params: 0
_________________________________________________________________
```

Fig.11. CNN Model

D. Hybrid CNN - LSTM

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 187, 64)           256
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 186, 64)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 186, 32)           6176
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 185, 32)           0
_________________________________________________________________
lstm (LSTM)                  (None, 185, 128)          82432
_________________________________________________________________
flatten (Flatten)            (None, 23680)             0
_________________________________________________________________
dense (Dense)                (None, 256)               6062336
_________________________________________________________________
dropout (Dropout)            (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 645
=================================================================
Total params: 6,184,741
Trainable params: 6,184,741
Non-trainable params: 0
_________________________________________________________________
```

Fig. 12. Hybrid CNN-LSTM Model

E. Multiclass Classification

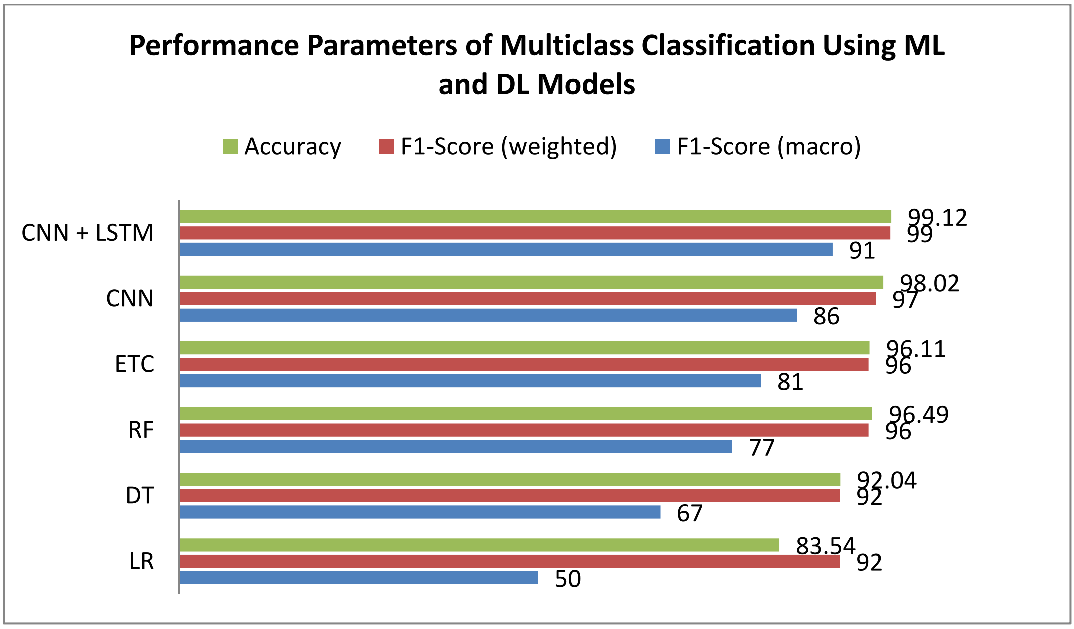

Fig. 13. Comparison of Performance Parameters of Multiclass Classification Using ML and DL Models

Table 2 and Figure 13, show the findings, and it is clear that the Hybrid CNN - LSTM model obtained the highest F1-Score (macro) of 91, demonstrating its accuracy in categorising ECG signals for important illness diagnosis. This model has a high F1-Score (weighted) of 99, which demonstrates how well it can handle unbalanced class distributions. The CNN + LSTM model's accuracy, which is 99.12%, further illustrates its exceptional performance. Comparatively speaking, the other algorithms likewise produced impressive outcomes. With F1-Scores (macro) of 77 and 81, respectively, and excellent accuracy rates of 96.49% and 96.11%, the Random Forest (RF) and Extra Trees Classifier (ETC) models performed well. With an F1-Score (macro) of 86 and an accuracy of 98.02%, the CNN model performed better. Lower F1-Scores for the Logistic Regression (LR) and Decision Tree (DT) models show a need for improvement. The F1-Score (weighted) accounts for class disparities by giving the minority classes a larger weight. The CNN + LSTM model fared better than all other algorithms in this regard, followed by the CNN, ETC, RF, DT, and LR models. These findings demonstrate the potency of deep learning models, especially the CNN + LSTM architecture, for analysing and identifying life-threatening disorders from ECG data. Convolutional neural networks and recurrent neural networks function well together because they can capture spatial and temporal relationships in the ECG data, which enhances performance.

Algorithms	**F1-Score (macro)**	**F1-Score (weighted)**	**Accuracy**
LR	50	92	83.54
DT	67	92	92.04
RF	77	96	96.49
ETC	81	96	96.11
CNN	86	97	98.02
CNN + LSTM	91	99	99.12

Table 2. Comparison of Performance Parameters of Multiclass Classification Using ML and DL Models

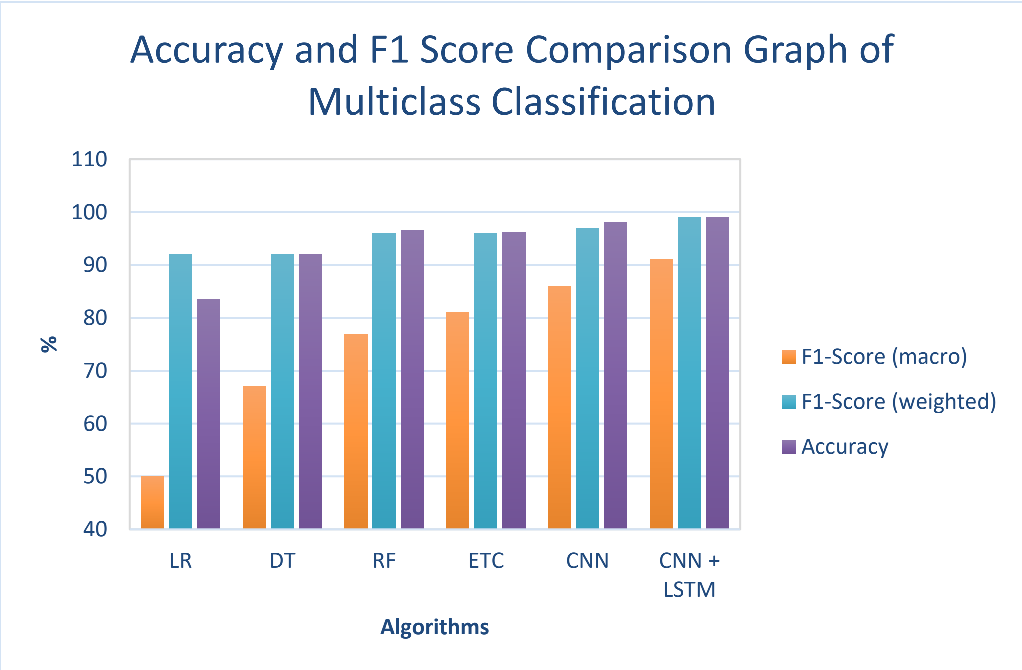

Fig. 14. Comparison of Performance Parameters of Multiclass Classification Using ML and DL Models

F. Binary Classification

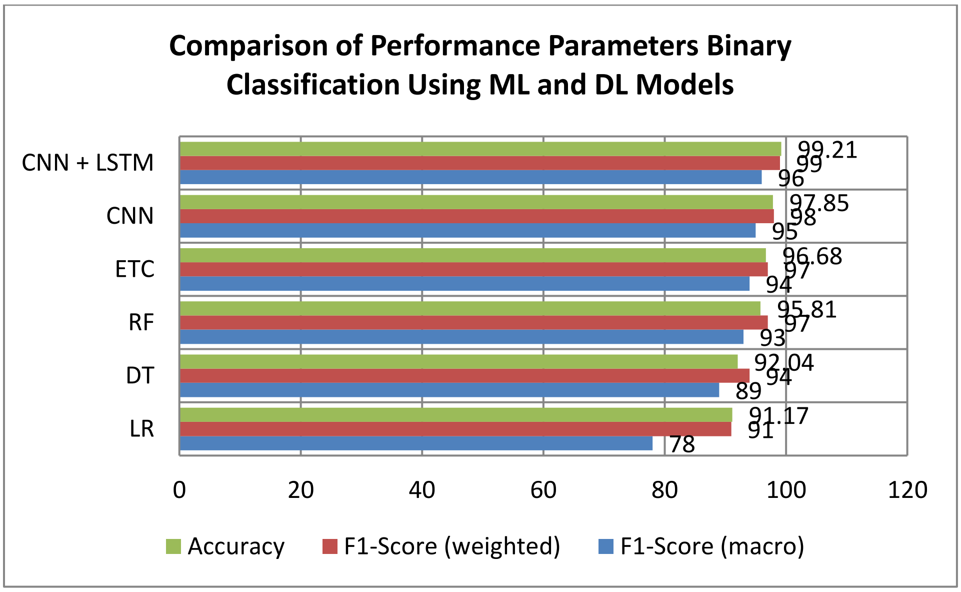

Fig. 15. Comparison of Performance Parameters Binary Classification Using ML and DL Models

It is clear from the findings in Table 3 and Figure 16 that all algorithms successfully completed the binary classification job. The model with the highest F1-Score (macro) of 96 and amazing accuracy of 99.21% was Hybrid CNN + LSTM. This demonstrates the model's accuracy in identifying critical and non-critical illnesses based on ECG data. With an F1-Score (macro) of 95 and an accuracy of 97.85%, the CNN model likewise performed quite well. High F1-Scores (macro) of 93 and 94 were attained by the Random Forest (RF) and Extra Trees Classifier (ETC) models, respectively, coupled with accuracy rates of 95.81% and 96.68%. These findings demonstrate the potency of ensemble learning techniques for categorising ECG data into two categories. While achieving reasonable results, the Logistic Regression (LR) and Decision Tree (DT) models exhibited somewhat lower F1-Scores, with F1-Scores (macro) of 78 and 89, and accuracies of 91.17% and 92.04%, respectively. Class imbalances are taken into consideration and a more thorough evaluation metric is provided by the F1-Score (weighted). The CNN + LSTM model fared better in this situation than any other algorithm, followed by the CNN, ETC, RF, DT, and LR models. The F1 scores and accuracy in the binary classification challenge show that machine learning algorithms, particularly deep learning models, are effective at correctly identifying and analysing important illnesses from ECG data. These results highlight the clinical applicability of these algorithms, such as the early identification and diagnosis of heart problems.

Algorithms	F1-Score (macro)	F1-Score (weighted)	Accuracy
LR	78	91	91.17
DT	89	94	92.04
RF	93	97	95.81
ETC	94	97	96.68
CNN	95	98	97.85
CNN + LSTM	96	99	99.21

Table 3. Comparison of Performance Parameters Binary Classification Using ML and DL Models

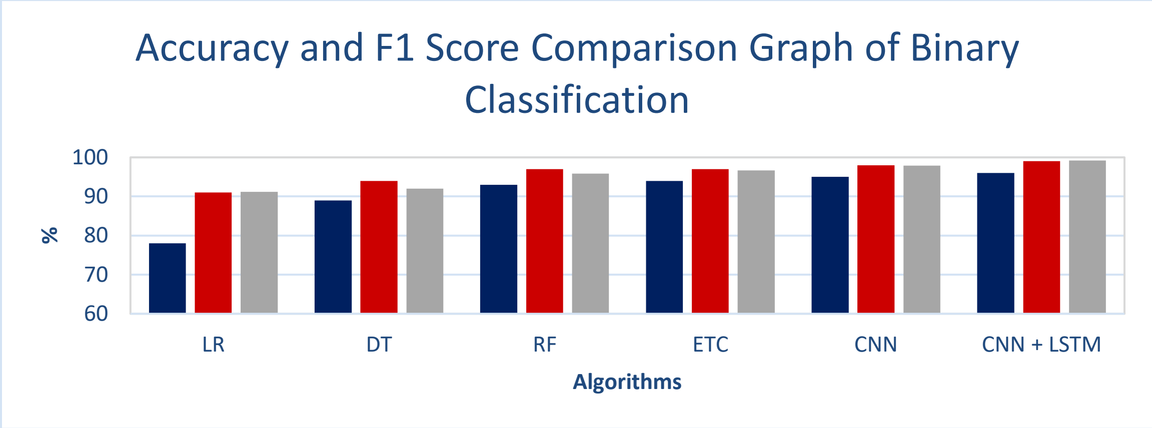

Fig. 16. Comparison of Performance Parameters Binary Classification Using ML and DL Models

Conclusion

As can be observed from the findings, the Hybrid CNN - LSTM model has the highest F1-Score (macro) of 91, demonstrating its accuracy in identifying ECG signals for important illness identification. This model demonstrates its versatility by having a high F1-Score (weighted) of 99. It can handle unequal class distributions. The CNN + LSTM model's 99.12% accuracy further exemplifies its exceptional performance. The other algorithms likewise produced impressive results in comparison. The F1-Scores (macro) of 77 and 81, respectively, for the Random Forest (RF) and Extra Trees Classifier (ETC) models, with respectable accuracy rates of 96.49% and 96.11%, were good. With an F1-Score (macro) of 86 and an accuracy of 98.02%, the CNN model performed better. Lower F1-Scores for the Logistic Regression (LR) and Decision Tree (DT) models suggest that they need to be improved. The F1-Score (weighted) provides the minority classes a larger weight to account for class inequalities. The CNN + LSTM model fared better than all other algorithms in this regard, followed by the CNN, ETC, RF, DT, and LR models. These studies show the efficiency of deep learning models, especially the Hybrid CNN - LSTM architecture, for evaluating and detecting life-threatening disorders using ECG data. Because they can detect spatial and temporal connections in the ECG data, convolutional and recurrent neural networks operate better together. The outcomes show that every algorithm successfully completed the binary classification challenge. The best F1-Score (macro) of 96 and incredible accuracy of 99.21% go to the CNN + LSTM model. This demonstrates how well the model does when utilising ECG data to categorise illnesses into critical and non-critical ones. With an F1-Score (macro) of 95 and an accuracy of 97.85%, the CNN model likewise did well. Models with high F1-Scores (macro) of 93 and 94 and accuracy rates of 95.81% and 96.68%, respectively, are Random Forest (RF) and Extra Trees Classifier (ETC). These findings demonstrate the potency of ensemble learning techniques for categorising ECG data into two categories. Although the F1-Scores (macro) of the Logistic Regression (LR) and Decision Tree (DT) models were slightly lower and their accuracies were 91.17% and 92.04%, respectively, they also delivered respectable results. The F1-Score (weighted) offers a more thorough evaluation metric while taking class imbalances into consideration. The CNN + LSTM model outperformed the CNN, ETC, RF, DT, and LR models in this instance. The binary classification challenge's accuracy and F1 scores show that deep learning models, in particular, are effective at correctly recognising and assessing important illnesses from ECG data. These results illustrate the potential clinical applications of these algorithms, including the early identification and diagnosis of heart problems.

References

[1] Zhang, J., Li, B., Xiang, K., Shi, X. (2019). Method of diagnosing heart disease based on deep learning ECG signal. arXiv preprint arXiv:1907.01514.

[2] Ouali, M.A., Ghanai, M., Chafaa, K. (2020). TLBO optimization algorithm based-type2 fuzzy adaptive filter for ECG signals denoising. Traitement du Signal, 37(4): 541-553. https://doi.org/10.18280/ts.370401

[3] Karpagachelvi, S., Arthanari, M., Sivakumar, M. (2010). ECG feature extraction techniques-a survey approach. arXiv preprint arXiv:1005.0957.

[4] Porumb, M., Iadanza, E., Massaro, S., Pecchia, L. (2020). A convolutional neural network approach to detect congestive heart failure. Biomedical Signal Processing and Control, 55: 101597. https://doi.org/10.1016/j.bspc.2019.101597

[5] Haberl, R.G.R.G., Jilge, G., Pulter, R., Steinbeck, G. (1989). Spectral mapping of the electrocardiogram with Fourier transform for identification of patients with sustained ventricular tachycardia and coronary artery disease. European Heart Journal, 10(4): 316-322. https://doi.org/10.1093/oxfordjournals.eurheartj.a05948 8

[6] Ebrahimi, Z., Loni, M., Daneshtalab, M., Gharehbaghi, A. (2020). A review on deep learning methods for ECG arrhythmia classification. Expert Systems with Applications: X, 7. 100033. https://doi.org/10.1016/j.eswax.2020.100033

[7] Jemilehin, T., Adu, M. (2016). Design and simulation of electrocardiogram circuit with automatic analysis of ECG signal. Analele Universitatii'Eftimie Murgu', 23(1): 155-172.

[8] Mougoufan à, J.B.B., Fouda, J.A.E., Tchuente, M., Koepf, W. (2020). Adaptive ECG beat classification by ordinal pattern based entropies. Communications in Nonlinear Science and Numerical Simulation, 84: 105156. https://doi.org/10.1016/j.cnsns.2019.105156

[9] Nurmaini, S., Darmawahyuni, A., Sakti Mukti, A.N., Rachmatullah, M.N., Firdaus, F., Tutuko, B. (2020). Deep learning-based stacked denoising and autoencoder for ECG heartbeat classification. Electronics, 9(1): 135. https://doi.org/10.3390/electronics9010135

[10] Mykoliuk, I., Jancarczyk, D., Karpinski, M., Kifer, V. (2018). Machine learning methods in electrocardiography classification. ACIT, 2018: 102-105.

[11] Zhang, W., Yu, L., Ye, L., Zhuang, W., Ma, F. (2018). ECG signal classification with deep learning for heart disease identification. In 2018 International Conference on Big Data and Artificial Intelligence (BDAI), Beijing, China, pp. 47-51. https://doi.org/10.1109/BDAI.2018.8546681

[12] Alarsan, F.I., Younes, M. (2019). Analysis and classification of heart diseases using heartbeat features and machine learning algorithms. Journal of Big Data, 6(1): 1-15. https://doi.org/10.1186/s40537-019-0244-x

[13] Alqudah, A.M., Qazan, S., Al-Ebbini, L., Alquran, H., Qasmieh, I.A. (2021). ECG heartbeat arrhythmias classification: a comparison study between different types of spectrum representation and convolutional neural networks architectures. Journal of Ambient Intelligence and Humanized Computing, 1-31. https://doi.org/10.1007/s12652-021-03247-0

[14] Ullah, A., Anwar, S.M., Bilal, M., Mehmood, R.M. (2020). Classification of arrhythmia by using deep learning with 2-D ECG spectral image representation. Remote Sensing, 12(10): 1685. https://doi.org/10.3390/rs12101685

[15] Pławiak, P., Acharya, U.R. (2020). Novel deep genetic ensemble of classifiers for arrhythmia detection using ECG signals. Neural Computing and Applications, 32(15): 11137-11161. https://doi.org/10.1007/s00521-018-03980-2

[16] Khetani, V. ., Gandhi, Y. ., Bhattacharya, S. ., Ajani, S. N. ., & Limkar, S. . (2023). Cross-Domain Analysis of ML and DL: Evaluating their Impact in Diverse Domains. *International Journal of Intelligent Systems and Applications in Engineering*, *11*(7s), 253–262.

[17] Khorrami, H., Moavenian, M. (2010). A comparative study of DWT, CWT and DCT transformations in ECG arrhythmias classification. Expert systems with Applications, 37(8): 5751-5757. https://doi.org/10.1016/j.eswa.2010.02.033

[18] Jenkal, W., Latif, R., Toumanari, A., Dliou, A., El BaAZcharri, O., Maoulainine, F.M.R. (2016). An efficient algorithm of ECG signal denoising using the adaptive dual threshold filter and the discrete wavelet transform. Biocybernetics and Biomedical Engineering, 36(3): 499-508. https://doi.org/10.1016/j.bbe.2016.04.001

[19] Nasim, A., Sbrollini, A., Morettini, M., Burattini, L. (2020). Extended segmented beat modulation method for cardiac beat

classification and electrocardiogram denoising. Electronics, 9(7): 1178. https://doi.org/10.3390/electronics9071178

[20] Liu, M., Kim, Y. (2018). Classification of heart diseases based on ECG signals using long short-term memory. 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, pp. 2707-2710. https://doi.org/10.1109/EMBC.2018.8512761

[21] Chawla, M.P.S. (2011). PCA and ICA processing methods for removal of artifacts and noise in electrocardiograms: A survey and comparison. Applied Soft Computing, 11(2): 2216-2226. https://doi.org/10.1016/j.asoc.2010.08.001

[22] Mr. R. Senthil Ganesh. (2019). Watermark Decoding Technique using Machine Learning for Intellectual Property Protection . International Journal of New Practices in Management and Engineering, 8(03), 01 - 09. https://doi.org/10.17762/ijnpme.v8i03.77

[23] Thompson, A., Walker, A., Rodriguez, C., Silva, D., & Castro, J. Machine Learning Approaches for Sentiment Analysis in Social Media. Kuwait Journal of Machine Learning, 1(4). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/153

[24] Kshirsagar, P. R., Reddy, D. H., Dhingra, M., Dhabliya, D., & Gupta, A. (2022). A review on comparative study of 4G, 5G and 6G networks. Paper presented at the Proceedings of 5th International Conference on Contemporary Computing and Informatics, IC3I 2022, 1830-1833. doi:10.1109/IC3I56241.2022.10073385 Retrieved from www.scopus.com